Dynamic Negative Sampling for Contrastive Recommendation

ABSTRACT

Negative sampling (NS) is an effective and efficient technique for representation learning models to obtain negative or unobserved instances. Particularly in the domain of recommendation systems (RSs), different NS approaches offer recommenders the ability to alleviate the data sparsity issue by sampling uninteracted with items using different strategies. Early pairwise ranking recommendation systems (e.g. BPR and its variants) only sample a single negative item for each positive item; a simple yet effective NS approach. Differently, some recent approaches select multiple negative items conditioned on the items' auxiliary information or generate hard negative representations for items using contrastive learning. However, these negative sampling approaches are either limited by the availability of the auxiliary information or unable to effectively search for informative negative items. In this paper, we propose a novel scheme, called Dynamic Negative Sampling (DNS), which can dynamically explore informative negative items and can generally improve the recommendation performance of embedding-based RSs. In addition, we adapt an existing objective function called Information Noise Contrastive Estimation (InfoNCE) to specifically leverage the informative negative items instead of the random ones used in InfoNCE. Furthermore, we design a novel approach to measure the entropy among the distributions of users and items. Using such an approach, we show how using our DNS scheme for sampling negative items can induce a higher information gain. Our extensive experiments on three public datasets demonstrate that our proposed DNS scheme can outperform existing state-of-the-art negative sampling approaches and can further enhance competitive recommender systems including LightGCN, SGL and SimGCL by upto 30.9% in terms of NDCG@10.

ACM Reference Format:

. 2022. Dynamic Negative Sampling for Contrastive Recommendation. In *Proceedings of 31st ACM International Conference on Information and Knowledge Management (CIKM '22).* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Recommender systems (RSs), which aim to provide items of interest to users from a large corpus of items, have achieved many successes in applications such as online shopping [27, 42], video recommendations [13, 31] and social platforms [26, 52]. Modern RSs usually need to be effectively and efficiently scaled up-to millions of items and active online users [10]. To achieve this, three different learning paradigms for modern RSs have been proposed, namely pointwise learning [11, 43], pairwise learning [36] and listwise learning [29, 60]. Among the three paradigms, pairwise learning has been generally recognised as the most effective and efficient one for handling large-scale and sparse implicit recommendation datasets [15, 16, 36, 46]. In particular, the Bayesian Personalised Ranking (BPR) model is one of the classic pairwise learning objective functions that can be adapted by either non-neural models [12] or neural models [15, 49], achieving state-of-the-art recommendation performances.

Under the training and optimisation process of BPR, each input instance is constructed by a triplet consisting of one user and a pair of positive & negative items, where the negative items are randomly sampled from the whole corpus of each user's uninteracted with items (i.e. random Negative Sampling (NS)). Ideally, all negative items need be used to construct the training triplets. However, given that users only interact with a few items, it is infeasible to pair each (user, positive item) with all negative items [47]. Hence, BPR and its variants use the random NS approach to assign a single negative item to each (user, positive item) pair. Although efficient, the random NS approach lacks the ability to search for more informative negative items, which are important for an enhanced recommendation performance [7]. Specifically, we consider those items that are unlikely to be interacted with by a user as the informative negative items for this user. More importantly, the random NS approach might induce false-negatives and the so-called exposure bias [22]. For example, when the interaction data are split into training/validation/test sets, some randomly sampled negative items for the training process might actually be positive items in the test set. These false-negative items will likely hurt the recommendation performance at the evaluation stage [14, 62]. In addition, the pairwise learning paradigm assumes that randomly sampled negative items are not preferred by the users, which might cause the exposure bias issue [4, 22] since these randomly sampled negative items might have never been presented to the user. To alleviate these issues of the random NS approach, some recent works have been proposed to modify the sampling strategy by considering the items' popularity [61] or the items' auxiliary information [22, 32] (e.g. geographical locations, tags). However, these NS approaches are either less effective or heavily rely on conditional items selection based on sparse categorisations of items. For example, a negative sampling approach that based on geographical information may fail to sample appropriate negative samples when the location is missing or inaccessible due to the privacy issue. Due to the fact that the categorisations of items are not always available, those NS approaches cannot be deployed in general interaction datasets. Therefore, a more generalised NS approach that does not rely on auxiliary information is highly desirable.

To alleviate the difficulty of choosing appropriate negative items without auxiliary information, we aim to incorporate the embeddings learned by a general recommender system. Given that the performance of a recommender system generally increases along the training process before saturation, one can become more and more confident that the lower-ranked items generated by the trained model are less likely to be interacted with by each user. Specifically,

CIKM '22, October 17-22, 2022, Atlanta, GE

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00

https://doi.org/10.1145/nnnnnnn.nnnnnnn

with the learned embeddings generated after each training epoch, we can better predict which item is an informative negative item in comparison to random sampling. Hence, our underlying intuition is that dynamically sampling negative items using the learned embeddings during the training process can help embedding-based recommender systems to fetch more informative negative samples. Indeed, our intuition also aligns with the core concept of importance sampling [1, 19, 21], where more informative samples could be derived during training so that the model's effectiveness can be enhanced with these samples.

In particular, we propose a general sampling scheme, called Dynamic Negative Sampling (DNS). Our DNS scheme can dynamically update negative items solely based the embeddings of users and items learned during the training of a general recommender without requiring any auxiliary information nor any extra data preprocessing. To effectively sample contrastive negative items, we propose a similarity-based method to dynamically search for contrastive negative items at each checkpoint. Furthermore, to leverage these additional negative items sampled at each checkpoint, our DNS scheme also includes an objective function motivated by the Information Noise Contrastive Estimation (InfoNCE) loss [34], a contrastive loss function originally devised to push multiple negative samples away from the positive sample. However, in contrast to InfoNCE, we design our objective function to specifically leverage the informative negative items instead of the random ones used in InfoNCE. Furthermore, based on information theory, we know that a distribution with a higher entropy contains richer information. To further motivate our proposed DNS scheme, we directly inspect the dynamic change of information during the training by introducing an entropy measure. Specifically, we use the proposed measure to approximate the information contained in different groups of negative items in order to compare the informativeness of random negative items with the dynamically sampled ones. Our obtained empirical results validate our intuition (i.e. performance improvements can be derived from a higher information gain) and echo a recent finding [59].

To summarise, we make the following contributions:

- We propose a similarity-based method to search for contrastive negative items.
- We propose a dynamic negative sampling (DNS) scheme, which can dynamically update the negative items and improve the recommendation performance of general RSs.
- We propose a novel contrastive objective function inspired by InfoNCE to work collaboratively within the DNS scheme in order to leverage multiple negative items.
- We propose an entropy measure to observe the change of information by approximating the dynamic information gain obtained from negative samples.
- Experiments on three public datasets demonstrate that our proposed scheme can significantly enhance the performance of existing RSs including LightGCN [15], SGL [49], SimGCL [56] and BPR [37] by up-to 30.9% and outperform strong baselines enhanced by other NS approaches.

2 RELATED WORK

In this section, we briefly introduce three bodies of related works: *negative sampling*, *contrastive learning* and *contrastive recommenders*.

2.1 Negative Sampling

Negative sampling (NS) approaches are widely applied in pairwise and listwise learning recommendation models to sample negative items for (user, positive item) pairs. Given that most of the items are negative items for each user, NS is the process of selecting a portion of negative items from the whole corpus of the user's uninteracted with items. In general, there are mainly two types of NS approaches, namely heuristic [32, 36] and model-based NS approaches [57]. The heuristic NS approaches set the sampling distribution according to some prior knowledge [32] or heuristic assumption [36], while the model-based NS approaches are based on the embeddings learned from models [57]. Since our proposed DNS scheme only leverages the learned embeddings, we classify it as a model-based approach.

Among all heuristic NS approaches, random NS is the most commonly used approach due to its simplicity. However, as mentioned in Section 1, random NS assumes that all items are presented to the users with the same probability, which might induce false-negative items [14, 62] or the exposure bias [4, 22]. To alleviate this exposure bias, the popularity-based NS approach [53] proposed to sample those popular items more often than cold items. However, such an approach might even exacerbate false-negatives [41]. In addition, some auxiliary information, such as the geographical information [32], social relations [32] and textual reviews [45], have also been leveraged by heuristic NS approaches to select those negative items that are unlikely to have been interacted with by a specific user. Although these heuristic NS approaches are efficient, we argue that they are either unable to be adapted to the dynamic learning of a recommendation model or difficult to generalise to general interaction dataset without auxiliary information. Therefore, in this paper, we aim to propose a general NS approach solely based on the embeddings learned from a recommender. Our proposed approach can be seen as a model-based NS approach.

Among different types of model-based NS approaches, generative adversarial networks (GAN)-based approaches [28, 35, 44, 55] have been shown to be effective. However, training an adversarial network is often time-consuming, making it hard to efficiently deliver precise recommendations to users [2]. Another model-based NS approach, known as dynamic NS [9, 47], uses the embeddings learned during the dynamic training of a recommendation system. Similar to [57], our proposed scheme includes a dynamic NS approach to update negative items. However, different from existing work [47, 57] using dynamic NS iteratively after each epoch, our DNS scheme is only activated after a number of epochs to improve the efficiency of the dynamic NS approaches. In addition, most of the existing works use the *in-batch* negatives [20, 54], meaning that their negative samples are selected from each training batch instead of being selected from all available samples. This *in-batch* negative sampling requires lower computational cost but it is suboptimal in terms of the effectiveness compared with sampling the global negatives from all available samples [51]. Hence, different from those NS approaches using the less effective *in-batch* negatives [20, 54], our proposed DNS scheme leverages the *global* negative items, meaning that we search among all negative items of each user instead of searching in a single batch of data.

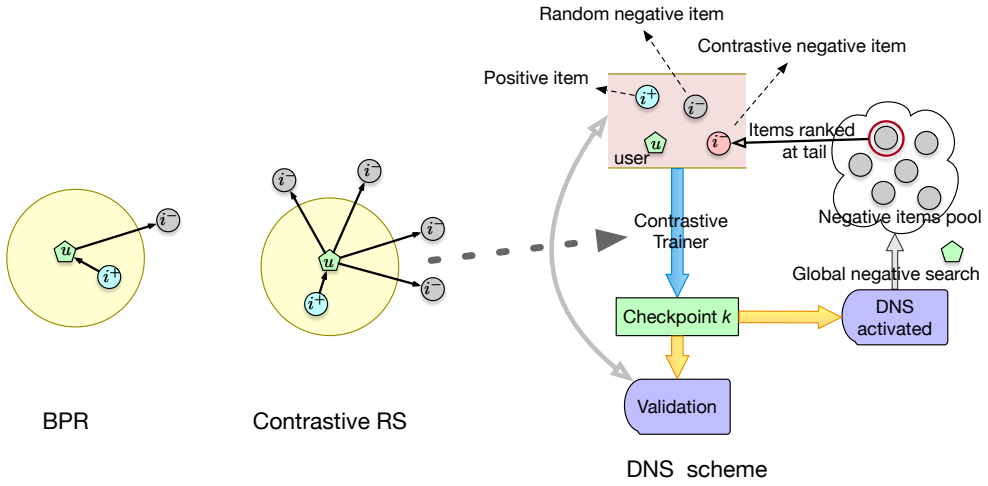

Figure 1: Comparison of our proposed DNS scheme with BPR.

2.2 Contrastive Learning

Contrastive learning is an emerging deep learning paradigm widely incorporated in the self-supervised learning method. Due to its promising ability to learn the representations of data without labels [7, 23], it is often deployed in various natural language processing (NLP) [50] and computer vision (CV) [5, 6] tasks, particularly when labels are sparse. Contrastive learning has been shown to yield state-of-the-art performances on different tasks by incorporating effective data augmentation techniques [39], hard negative sampling [38] and contrastive objective functions e.g. InfoNCE [34]. For example, SimCLR [5], a contrastive framework, has been demonstrated to be even more effective on the image classification task than some supervised models. In information retrieval, a state-of-the-art dense retrieval model called ANCE [51] has been proposed using the hard negative sampling technique to select negative documents easily recognised as positive documents by a dense retriever. Pairwise learning is one of the most popular learning paradigm in the recommendation systems field [36], where its goal is to learn representations for available samples such that similar samples stay close to each others, which is analogous to the goal of contrastive learning. The main difference is that contrastive learning usually involves only one sample type, such as all image samples, while pairwise recommenders have two types of samples, namely users and items. Moreover, a contrastive objective function can usually handle multiple negative samples, while pairwise recommenders only sample a single negative item for each training instance.

2.3 Contrastive Recommender Systems

Contrastive recommender systems are broadly defined as those recommender systems using contrastive learning techniques including the corresponding data augmentations [18] and sampling techniques [62] as well as the contrastive loss [18, 30, 49]. Data augmentation techniques are usually used to generate synthesised embeddings for contrastive learning. Specifically, contrastive recommenders can generate synthesised embeddings for both users and items either by summing up embeddings from the pre-defined neighbourhood of a user or an item [18] or by randomly masking out some elements from the learned embeddings [49]. Although these data augmentation techniques have been shown to be effective in the field of CV and NLP, a previous study [56] has theoretically proven that they cannot enhance the performance of RSs without the contrastive loss. Intuitively, it is reasonable to teach a CV or NLP model to predict a rotated/cropped image [6] or a partially masked sentence [8] with the same label or semantic meaning as the original image or sentence. However, such embedding augmentation and random dropout techniques used by contrastive recommenders such as MixGCF [18] and SGL [49] cannot provide a semantic information for those synthesised embeddings because users and items that correspond to those synthesised embeddings do not exist at all. Therefore, different from these contrastive recommenders relying on synthesised embeddings, our DNS scheme searches for contrastive negative items from the existing items. Existing contrastive recommenders have incorporated different types of contrastive loss functions, but most of them only partially use the loss by summing it up with the original pairwise loss. By doing so, the contrastive loss can only be regarded as a regularisation. Hence, an additional regularisation parameter is also required to control its weight. Distinctively, our proposed scheme can fully exploit the effectiveness of the contrastive loss by discarding the commonly used pairwise loss.

In this paper, we propose a novel DNS scheme including a dynamic negative sampling approach and a loss function inspired by InfoNCE. Differently from existing works, our DNS scheme can explicitly search for contrastive samples and update them in a dynamic manner. Meanwhile, our proposed loss function is specifically tailored for contrastive recommendation.

3 METHODOLOGY

In this section, we first introduce the basic settings and the corresponding notations in pairwise learning recommender systems (Section 3.1). Next, we define contrastive recommendation and its

relationship with BPR (Section 3.2), followed by our dynamic negative sampling approach (Section 3.3) used by DNS. Finally, we demonstrate how to incorporate our proposed scheme and objective function for a general recommender system (Section 3.4). To provide a clear overview of our proposed scheme, we use Figure 1 to illustrate the architecture of DNS in comparison with BPR.

3.1 Notations

We consider a recommender system with a users set $\mathcal{U}$ ($|\mathcal{U}| = M$) and an items set $\mathcal{I}$ ($|\mathcal{I}| = N$). Let the entire set of implicit feedback $\mathbf{R} \in \mathbb{R}^{M \times N}$ be the binary matrix with entries being 1/0 representing the positive/negative historical interactions between users and items. Our task is to rank relevant items for each user given their historical interactions. We here consider implicit feedback instead of the explicit rating feedback because of the abundance of the former. We denote $\mathbf{R}_{ui} = 1$ if the user u has interacted with the item i, otherwise $\mathbf{R}_{ui} = 0$. From the matrix $\mathbf{R}$, we can generate our training set $\boldsymbol{D} : \mathcal{U} \times \mathcal{I}^+ \times \mathcal{I}^-$, where the former $\mathcal{I}^+$ is the set of positive items and the latter $\mathcal{I}^-$ is the set of negative items. We use i_u^+, i_u^-, and $i_u^-{}'$ to denote a positive item, a negative item and a contrastive negative item of a user u.

3.2 Contrastive Recommendation

Contrastive recommenders [30, 49, 56] are designed to learn contrastive representations for users and items such that the distance between users and their positive items are minimised while the distance between users and their negative items are maximised. Among existing contrastive recommenders, InfoNCE is the most commonly adopted objective function [5]. In general, contrastive recommenders optimise the following InfoNCE objective function:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp\left(\text{sim}(\mathbf{e}_u, \mathbf{e}_{i_u^+})/\tau\right)}{\exp\left(\text{sim}(\mathbf{e}_u, \mathbf{e}_{i_u^+})/\tau\right) + \sum_1^j \exp\left(\text{sim}(\mathbf{e}_u, \mathbf{e}_{i_u^-})/\tau\right)}, \tag{1}$$

where $\text{sim}(\cdot)$ is a similarity measure and τ is a constant denoting the *temperature parameter*; $\mathbf{e}_u$, $\mathbf{e}_{i_u^+}$ and $\mathbf{e}_{i_u^-}$ represent the embeddings of user u and his/her interacted and uninteracted with items, respectively.

In Equation (1), we note that there are j different negative samples. In the existing literature [30, 62], such negative samples are obtained by leveraging different data augmentation methods. For example, the graph perturbation has been used to sample negative samples from the multi-hop graph neighbourhood [30]. Besides, the node dropout, edge dropout, and random walk approaches have also been used to generate negative samples [49]. Given that existing contrastive RSs generally follow a similar loss function, we can conclude that the approach used to generate the negative samples is the underlying factor in differentiating contrastive recommenders.

Indeed, we can bridge the BPR pairwise ranking approach with contrastive recommenders by defining the objective function of BPR in a similar fashion. For example, given each training instance of BPR constructed as:

$$D_u := \{\langle i_u^+, i_u^- \rangle \,|\, i_u^+ \in \mathcal{I}_u^+ \wedge i_u^- \in \mathcal{I} \setminus \mathcal{I}_u^+\}, \tag{2}$$

a BPR recommender optimises the following objective function:

$$\mathcal{L}_{\text{BPR}} = \sum_{(u,i^+,i^-)\in \boldsymbol{D}} -\log \frac{\exp(\mathbf{e}_u \cdot \mathbf{e}_{i_u^+})}{\exp(\mathbf{e}_u \cdot \mathbf{e}_{i_u^-}) + \exp(\mathbf{e}_u \cdot \mathbf{e}_{i_u^+})}, \tag{3}$$

where λ is the regularisation parameter; θ represents all parameter embeddings in the trained model.

By comparing Equation (1) and Equation (3), we notice that there are two main differences: 1) BPR uses $\exp(\cdot)$ to model the interaction between users and items while the InfoNCE loss uses $\exp(\cdot)$ together with a similarity measure; 2) in the BPR setup, only a static randomly sampled negative item is incorporated for each training instance, while for contrastive recommenders, multiple negative items are incorporated. A detailed comparison between BPR with a contrastive recommender is illustrated in Figure 1. Since both approaches aim to repel negative items away from users and positive items, we can consider a contrastive recommender as a more generalised extension of BPR with additional negative items. The difference between contrastive recommenders and BPR is that contrastive recommenders use different data augmentation methods to create or search for more informative negative samples.

Interestingly, a recent study [56] has theoretically demonstrated that the existing data augmentation methods, which are used by contrastive recommenders cannot provide robust performance improvements. In their study [56], the authors have shown that the data augmentation methods, including node dropout and random walk, cannot significantly improve the performance of a contrastive recommender performance, with the variant without augmentations sometimes performing better. We argue that existing models are limited by their statically sampled negative items. Although those data augmentation methods can effectively create different variants of embeddings (known as different views in contrastive learning) to serve as contrastive negatives, these variants are still rooted in the same user/item. Hence, the whole process can only be treated as static because they keep a constant sampling distribution across the training process.

Therefore, to further progress the development of contrastive recommenders, we propose to sample contrastive negative items dynamically. During the dynamic sampling process, we expect the model to gradually learn which negative items are more informative than random negative items. Thus, our contrastive negative items are updated accordingly during this learning process. In the following section, we formally define our proposed Dynamic Negative Sampling (DNS) scheme by detailing how to search for contrastive negative items and the subsequent loss function.

3.3 Dynamic Negative Sampling

As mentioned in Section 1, using the static approach (e.g. the one used in BPR) to randomly sample negative items might lead to an increase of false-negatives and can result in the exposure bias issue. We propose to use a dynamic approach instead of the existing static one and its variants [15, 32, 46].

The amount of each user u's negative items (i.e. $|\mathcal{I} \setminus \mathcal{I}_u^+|$) is large since $|\mathcal{I}| \gg |\mathcal{I}_u^+|$. That's the reason why a ideal sampling strategy is required to handle the huge amount of negative items because otherwise the size of the training set $\boldsymbol{D}$ will become a bottleneck. Although the random sampling strategy adopted by the pairwise

ranking approach is simple generally effective, these randomly sampled items can accidentally hurt the recommendation performance. For example, those randomly sampled negative items might actually be positive items in the test set, since the sampling is suppose to be conducted solely based on the training set $\boldsymbol{D}$. These false-negative items will likely degrade the recommendation performance. Besides, users might be paired with those negative items that have not been exposed to them. This might introduce a bias against unpopular items by assuming they are not preferred by users.

To alleviate the aforementioned issues of the random negative sampling, we devise a novel dynamic negative sampling scheme, abbreviated as DNS. Our DNS scheme can sample contrastive negative items and reduce the false-negatives simultaneously. Specifically, the DNS scheme can automatically update negative items at each checkpoint, solely relying on our proposed similarity-based method defined later, without any auxiliary information about the users and items.

Following existing contrastive models [5, 6, 38, 51], we use the cosine similarity to measure the similarity between each user and his/her negative items. In the field of CV and NLP, a negative sample is usually defined as a contrastive negative when it has a higher similarity score with the anchor point. This is a commonly used definition for most of the contrastive models [5, 38]. Distinctively, our DNS scheme defines contrastive negative items as those items, which have lower similarity scores with the target user because items with higher similarity scores are likely to be the items of the user's interest. As a consequence, using the exact definition of hard negatives from other fields might actually increase the false-negatives in a recommender system framework instead of benefiting the representation learning and the prediction performance. Our choice of dissimilar negative items will be further justified in the subsequent experiments. We use the following equation to search for the contrastive negative items:

$$i_u^{-\prime} \in f_{\text{topk}}\left(-\cos(\mathbf{e}_u, \mathbf{E}_{I_u^-})\right), \tag{4}$$

where $i_u^{-\prime}$ denotes the contrastive negative item for user u; $f_{\text{topk}}(\cdot)$ means that k elements with higher similarity scores will be retrieved ; $\cos(\cdot)$ is the cosine similarity; $\mathbf{E}_{I_u^-}$ is the embedding table containing the embeddings of all negative items for user u.

After sampling k contrastive negative items for each user, our DNS scheme will randomly assign one or more $i_u^{-\prime}$ for each user. Note that if a user has m positive items in the training set $\boldsymbol{D}$, m contrastive negative items $i_u^{-\prime}$ will be assigned to this user correspondingly. Therefore, the new training set $\boldsymbol{D'}$ will have the same number of training instances as the original set $\boldsymbol{D}$ but with the additionally sampled negative items in each instance.

For training the recommender, we proposed to leverage both statically and dynamically sampled negative items to combine the benefit of both the static and dynamic negatives during contrastive learning. In particular, our contrastive recommender aims to optimise the following objective function:

$$\mathcal{L}_{\text{con}} = \sum_{(u,i^+,i^-,i^{-\prime})\in D'} -\log \frac{f_{\text{s}}(\mathbf{e}_u, \mathbf{e}_{i_u^+})}{f_{\text{s}}(\mathbf{e}_u, \mathbf{e}_{i_u^-}) + f_{\text{s}}(\mathbf{e}_u, \mathbf{e}_{i_u^{-\prime}}) + f_{\text{s}}(\mathbf{e}_u, \mathbf{e}_{i_u^+})}, \tag{5}$$

where $f_{\text{s}} = \exp(\cos(\cdot)/\tau)$; the cosine similarity $\cos(\cdot)$ can also be replaced by other similarity measures but as mentioned above we use $\cos(\cdot)$ following the default setup of contrastive learning; $\mathbf{e}_u$, $\mathbf{e}_{i_u^+}$ and $\mathbf{e}_{i_u^-}$ have been defined under Equation (1), while $\mathbf{e}_{i_u^{-\prime}}$ is the embedding of a contrastive item.

By using the contrastive objective function $\mathcal{L}_{\text{con}}$, we aim to repel not only the static negative items but also the contrastive negative items away from each user and his/her corresponding positive items in the latent space. Indeed, by comparing Equation (5) and Equation (3), we can also interpret $\mathcal{L}_{\text{con}}$ as an extended version of $\mathcal{L}_{\text{BPR}}$, where $\mathcal{L}_{\text{BPR}}$ is dedicated to leverage the single negative case.

Since retrieving contrastive negative items will lead to a lower efficiency, we only update contrastive negative items iteratively after a fixed number of epochs to accelerate the overall training process. Specifically, we retrieve k contrastive negative items for each user at every checkpoint. Next, these updated contrastive negative items will be stored and used during the following n epochs till the next checkpoint. In addition, contrastive negative items from previous checkpoints will be dropped for memory-efficiency reasons.

3.4 Model Prediction

When incorporating the proposed DNS scheme into a general model, an existing recommender system still retains its original model initialisation and no additional data pre-processing is required. Our DNS scheme is activated only at each checkpoint during the training process according to a pre-defined update interval n. Therefor, our proposed scheme is straightforward enough to be incorporated by general RSs. Furthermore, our contrastive objective function $\mathcal{L}_{\text{con}}$ does not change the prediction layer, so no extra modification is required for the recommendation generation. In particular, for our used general RSs, namely BPR, LightGCN, SGL and SimGCL, we still use the dot product (BPR) or the dot product with their corresponding activation layers (LightGCN, SGL and SimGCL) as their prediction functions. Therefore, the target of general RSs and their DNS-based variants is to highly rank items preferred by users according to each model's prediction function.

3.5 Entropy Analysis

In information theory, entropy is the measure of a variable's average level of uncertainty or its informational value [3]. In fact, entropy is strongly related to many critical concepts of deep learning. For example, the cross-entropy loss [33, 58] is widely used for classification tasks. Moreover, information bottleneck [40], mutual information [24] and Kullback-Leibler divergence [17] can all be considered as a comparison of entropy between the observed and predicted distributions.

In this paper, we aim to determine how informative the negative items sampled by DNS are for a recommender system. In particular, to show the added-value of our DNS scheme, we need to compute such an informational value and compare the value of DNS-sampled negative items against randomly sampled negative items. Therefore, inspired by information theory, we leverage the entropy measure. Since the embeddings of users and items are of the same length and trained in the same latent space, we can roughly assume that they are sampled from the same distribution. Thus, to measure the entropy of a certain group of items and users, we propose to merge

the embeddings of these users and items and compute the entropy as follows:

$$H = -\frac{\text{sum}\Big(\sigma(\text{concat}(\mathbf{E}_u, \mathbf{E}_i)) \cdot \log(\sigma(\text{concat}(\mathbf{E}_u, \mathbf{E}_i)))\Big)}{d}, \quad (6)$$

where $\sigma(\cdot)$ is the softmax function used to normalise all embeddings to positive values to avoid computing a log of negative values; concat$(\cdot)$ stands for the concatenation operation of embeddings; d is the latent dimension of the users and items' embeddings i.e. $\mathbf{E}_u$ and $\mathbf{E}_i$, respectively.

Using Equation (6), we can compute the entropy of the users and items' embeddings. Our proposed entropy measure aims to approximate the information or uncertainty encapsulated in the sampled negative items. To validate our measure, we also examine it along with the performances of the examined models on the validation set during training. In principle, when the entropy measure saturates, the validation performance should also stabilise because there is no more information to be learned.

Table 1: Statistics of our used datasets.

	MovieLens	Yelp	Amazon
Users	6,038	19,539	21,596
Items	3,533	22,228	11,167
Interactions	575,281	450,884	2,092,329
Density	2.697%	0.104%	0.868%

4 DATASETS AND EXPERIMENTAL SETUP

We use three public datasets, i.e. MovieLens-1M[1], Yelp[2] and Amazon-instant-video[3], to evaluate our proposed DNS scheme. In particular, MovieLens-1M is a dataset containing interactions between users and movies; Yelp is a dataset of venue check-ins; and Amazon-instant-video is a subset of the Amazon review dataset. For the rest of the paper, we use 'MovieLens' and 'Amazon' as shorthands for 'MovieLens-1M' and 'Amazon-instant-video', respectively. Table 1 provides the statistics of the three used datasets. In the following, we aim to answer the following research questions:

- **RQ1** Do dissimilar negative items lead to a lower false-negative rate and enhanced recommendation performances?
- **RQ2** Can our proposed DNS scheme generally improve the recommendation performances for existing embedding-based RSs and outperform state-of-the-art NS-enhanced approaches?
- **RQ3** How do the introduced hyperparameters affect the effectiveness of the Dynamic Negative Sampling technique?
- **RQ4** Can negative items sampled by DNS provide a higher information gain than those random ones according to our proposed entropy measure?

In the next sections, we describe four general baselines and two state-of-the-art NS-enhanced approaches that we use to evaluate the performance of our proposed DNS scheme, followed by the used evaluation methodology, and the corresponding experimental setup.

[1] https://grouplens.org/datasets/movielens/ [2] https://www.yelp.com/dataset
[3] https://jmcauley.ucsd.edu/data/amazon/

4.1 Baselines

We evaluate the effectiveness of our DNS scheme on four general recommendation models (i.e. BPR, LightGCN SGL and SimGCL), which are also used as our baselines:

- **BPR** [36]: This is a conventional matrix factorisation model optimised by the pairwise ranking-based method using the random NS approach. Following the implementation described in [37], we also use the user bias, item bias and global bias.
- **LightGCN** [15]: Building upon NGCF [46], LightGCN is a more effective and efficient GNN-based neural recommender system with less redundant neural components. LightGCN uses a simplified GNN [48] to aggregate node embeddings from neighbourhoods.
- **SGL** [49]: SGL uses self-supervised learning including the node dropout, edge dropout and random walk augmentation techniques to generate multiple representations of users and items based on the structure of the graph. In addition, SGL has the ability to explore hard negative samples.
- **SimGCL** [56]: This is a recently proposed graph contrastive recommender with a high effectiveness and flexibility. SimGCL can enhance the contrastive representations of users and items by introducing a regulated noise sampled from the uniform distribution instead of relying on any graph augmentation techniques.

In addition, we compare DNS with two other state-of-the-art NS-enhanced approaches to demonstrate the superiority of our proposed scheme:

- **MixGCF** [18]: MixGCF is a recent general negative sampling (NS) approach that can be directly used to train GNN-based recommender systems. It creates synthetic negative samples through the idea of neighbourhood mixing. Among all of its variants combined with LightGCN, NGCF and PinSage, the LightGCN variant consistently outperforms the other two variants on all of the benchmark datasets with a large margin (at least 40% in terms of the recall@20 measure). Therefore, we only compare DNS with the best variant (i.e. MixGCF+LightGCN) and we use MixGCF to denote this best variant for the rest of the paper.
- **PRIS** [28]: Personalised Ranking loss based on Importance Sampling (PRIS) is an effective recommender system for items recommendation. It incorporates a novel NS approach based on importance sampling, where PRIS can assign larger weights to more informative negative items.

4.2 Evaluation Methodology

Following the commonly used evaluation setup for these datasets [15, 49], we split each dataset into a training set, validation set and test set with a ratio of 8:1:1. However, different from prior works that only use one oracle testing set per dataset with the sampled negative items, we construct 10 different testing sets with different sampled negative items for each dataset using different random seeds. This allows to reduce the evaluation bias on some specific testing negatives [25]. For a fair comparison, we keep the batch size as 1000 for all models and each model is trained by up-to 500 epochs unless it is early-stopped when the validation performance

Table 2: Performances and false-negative rates of BPR when using a static random sampler and three dynamic samplers on the three used datasets. The NDCG metric is computed at rank cutoff 10. F-N rate denotes the false-negative rate and * denotes a significant difference between the performance of the dynamic dissimilar sampler and that of other samplers according to the paired t-test with the Holm-Bonferroni correction for p <0.01. The best result and the lowest F-N rate are highlighted in bold.

Sampler	MovieLens		Yelp		Amazon	
	F-N Rate (%)	NDCG	F-N (%) Rate	NDCG	F-N Rate (%)	NDCG
Static random sampler	0.5139	0.2805*	0.1289	0.0887*	0.6001	0.0834*
Dynamic random sampler	0.5125	0.2910*	0.1201	0.0889*	0.604	0.0820*
Dynamic similar sampler	1.0562	0.2013*	0.2884	0.0656*	1.1796	0.0633*
Dynamic dissimilar sampler	**0.4853**	**0.3394**	**0.1033**	**0.1513**	**0.5787**	**0.1092**

does not increase for 50 successive epochs. To tune all hyperparameters, we apply a grid search, where the learning rate is tuned in $\{10^{-2}, 10^{-3}, 10^{-4}\}$; the latent dimension in $\{32, 64, 128\}$ and the L_2 normalisation in $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$.

To answer **RQ1**, recall that in Section 3.3 we defined the dissimilar negative items to be the contrastive negative items. Similarly, we can also follow ANCE and other contrastive models to define a similar negative item as a contrastive sample, which is commonly known as the hard negative. Besides, we also include a naive baseline sampler that dynamically samples random negative items. To justify our choice of dissimilar negatives, we compare the recommendation performance of DNS when adopting different types of negative items, namely dynamic dissimilar negatives, dynamic similar negatives and dynamic random negatives. In addition, we calculate the corresponding false-negative rates for all the aforementioned three cases, to further illustrate why choosing the dynamic dissimilar negatives is more appropriate in our recommendation scenario. The false-negative rate is computed by dividing the number of training instances whose negative items are false negatives according to the testing set, by the total number of training instances.

To answer **RQ2**, we need to validate the effectiveness of our proposed DNS scheme over two groups of comparative experiments, namely comparing our DNS-enhanced recommenders with their corresponding baseline recommenders and comparing our DNS-enhanced recommenders with other NS-enhanced recommenders. Three metrics (i.e. NDCG@10, Recall@10 and MAP@10) are applied to evaluate the effectiveness of all evaluated models and baselines.

Note that our DNS scheme contains two new hyperparameters, i.e. n and k, which are the update interval and the number of retrieved negative items at each checkpoint, respectively. To answer **RQ3**, we tune n and k from $\{1, 5, 10, 20, 30, 40, 50\}$ and $\{10, 50, 100, 150, 200, 250, 300\}$, respectively. We evaluate the recommendation performance of the BPR model with our DNS scheme. When $n = 1$, the trained model will have its negative items refreshed after every epoch and we denote this case as the exhaustive update.

Finally, to answer **RQ4**, after each training epoch, we calculate the entropy measure of BPR and BPR_{DNS}, respectively. For a fair comparison, we explicitly measure the entropy of contrastive negative items only for BPR_{DNS} to exclude the possible bias of having additional negative items over BPR. Meanwhile, we report the NDCG@10 performance of these two models in the same plot to demonstrate the validity of our proposed measure. For the latter case, we disable the early-stopping strategy to illustrate the whole training curve.

For both **RQ3** and **RQ4**, we only show the results of BPR and BPR_{DNS} but note that the experiments with other baselines result in the same conclusions (but we omit them here due to space limitations). Moreover, for **RQ4**, we need to report the validation performance iteratively after every epoch, which is time-consuming if deep neural models are used. Therefore, we also choose these two models for both efficiency reasons and space constraints, since they sufficiently illustrate the added-value of our proposed DNS scheme.

5 RESULTS ANALYSIS

In this section, we report the experimental results and answer our four research questions.

RQ1: Contrastive Negative Items. In order to answer **RQ1**, we compare three possible dynamic negative samplers with the static random sampler i.e. a plain BPR on the three used datasets. First, from Table 2, we find that there is no noticeable difference between the static random negative sampler and the dynamic random negative sampler in terms of the false-negative rate and NDCG@10. This suggests that solely refreshing negatives won't help in decreasing the false-negative rate and improving the recommendation performance. Second, by comparing the static random sampler and the dynamic similar sampler, we observe a significant performance decrease when using the similar negative sampler across all datasets. Furthermore, the false-negative rate is almost doubled on each dataset when the dynamic similar negative sampler is used, compared with the static BPR. This is because the dynamic similar negative sampler always selects highly ranked negative items during the training process. Although these highly ranked negative items might be the typical hard negatives, they are more likely to be the users' interacted with items, which leads to a higher false-negative rate. Alternatively, the dynamic similar negative sampler will sample some interesting items to the users that have not been presented to them yet, leading to an exposure bias. Therefore, sampling negative items with high similarity scores is not appropriate for RSs and it can hurt the recommendation performance by boosting the false-negative rate. On the contrary, the dynamic dissimilar negative sampler, which is used by DNS can consistently and significantly outperform all other negative samplers across all used datasets with a lower false-negative rate.

As a summary, our reported results suggest that contrastive negative items should be defined as those negative items with lower similarity scores with users, which indeed corroborates our choice of dynamic dissimilar negative sampler for DNS as discussed

in Section 3.3. In answer to **RQ1**, dissimilar negative items do lead to a lower false-negative rate and enhanced recommendation performances.

RQ2: Effectiveness of DNS. From Table 3, we can see that our four used baselines are improved with large performance margins compared to their DNS variants across all used metrics and datasets. This shows that our proposed DNS scheme can remarkably improve the recommendation performance of the general RSs. Noticeably, our BPR_{DNS} model can even outperform both NS-enhanced approaches (MixGCF, PRIS) on the MovieLens dataset. This suggests that the performance of a classic non-neural embedding-based recommender system can be largely boosted and can even outperform state-of-the-art neural recommenders when negative items of high quality are provided. Among all DNS-based models, the $\text{SimGCL}_{\text{DNS}}$ variant is shown to achieve the best performance for the most of cases. We also find that a better performing general recommender usually gives a better DNS variant. For example, SGL performs the best among all general recommenders on the MovieLens dataset, while SimGCL performs the best on the Yelp and Amazon datasets. As a result, their DNS variants achieve the overall best performances on the corresponding datasets. This is expected because a better performing general recommender can also provide negative items that are unlikely to be interacted with by users with a higher probability, compared with those worse performing baseline recommenders.

Overall, in answer to **RQ2**, our results show that our proposed DNS scheme can generally improve the recommendation performances of four embedding-based general recommenders with a large margin. Meanwhile, most of our DNS-based recommenders can outperform strong baselines including MixGCF and PRIS, which use other advanced NS approaches.

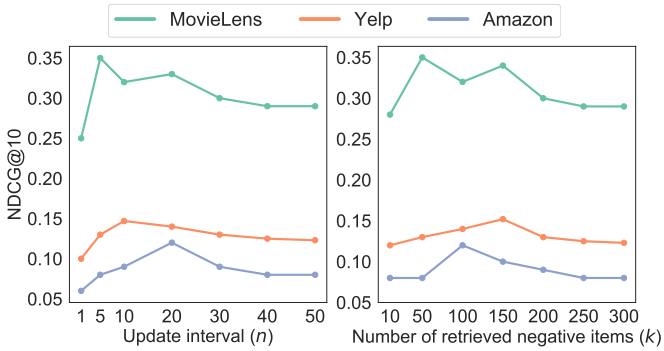

Figure 2: NDCG@10 performances of the BPR_{DNS} model over (a) different update intervals (n) and (b) different number of retrieved negative items (k) on three used datasets.

RQ3: Hyperparameter Analysis. To answer **RQ3**, recall that our proposed DNS scheme introduces two hyperparameters, namely the update interval (n) and the number of retrieved negative items (k), where n defines how frequently DNS will be activated to update the negative items and k controls the number of negative items to be retrieved for each user. We plot these two hyperparameters versus the NDCG@10 performance of the BPR_{DNS} model on three used datasets, where n varies from $\{1, 5, 10, 20, 30, 40, 50\}$ and k is tuned within $\{10, 50, 100, 150, 200, 250, 300\}$. We choose the maximum of n as 50 because we train all models with an early-stopping strategy, where each model will be early-stopped if its validation performance is not improved within 50 successive epochs. Hence, DNS might not be activated at all when n is larger than 50. k is tuned with a maximum of 300 because when the number of retrieved negative items is too large, these sampled negative items will tend to be the same as the random items. From Figure 2 (a), we can see that at the exhaustive update i.e. n=1, BPR_{DNS} will underperfom the plain BPR model for all datasets (NDCG@10=0.2479, 0.0987 and 0.0611 for the MovieLens, Yelp and Amazon datasets, respectively). This indicates that updating negative items too frequently will in turn hurt the recommendation performance of the general recommender. Indeed, training RSs on a relatively large dataset almost always takes more than 1 epoch to converge, hence this performance decline is due to the updating of the negative items with a high frequency. We also find that BPR_{DNS} peaks at n= 5, 10 and 20 for the MovieLens, Yelp and Amazon datasets, respectively. BPR_{DNS} peaks with a large n value on the Amazon dataset because the Amazon dataset has far more interactions than the other two datasets (e.g., about 4.5× size of the Yelp dataset), hence the longer convergence period is also expected, which means a frequent update is not desired. From Figure 2 (b), we find that sampling between 50 to 150 contrastive negative items help the BPR_{DNS} model to achieve the best performance on the three used datasets. When only 10 contrastive negative items are sampled, DNS is likely to cause repeated samples. For example, if a user has more than 10 interactions in the training set, repeated samples are unavoidable since only 10 contrastive negative items are selected. These repeated samples might degrade the recommendation performance. On the other hand, if a large amount of negative items are sampled, these sampled items tend to become as informative as those randomly sampled negative items. For example, when the top 1,000 negative items are sampled by DNS, the probability that each item is selected is $\frac{1}{1000} = 0.001$. This is not substantially larger than randomly sampling negative items from the MovieLens dataset (3,533 items in total) if we exclude all positive items. Furthermore, the effectiveness of DNS lies in that it can explore the users' actual negative items and estimate what these users dislike. Therefore, sampling from a large pool of negative items is not far from a random guess since not many users have a large amount of items they dislike.

RQ4: Entropy Analysis. To answer **RQ4**, we plot the entropy measure of the contrastive negative items (BPR_{DNS}) against the random negative items (BPR) together with the performances of two models on the validation set. First, we want to examine the validity of our proposed measure. We hypothesise that entropy can approximate the amount of information contained in the embeddings. From Figure 3, we can observe that when the entropy of the two models stabilises after epoch 50, the validation performances also stop fluctuating. The proposed measure is not an instant indicator because although we can approximate the level of information, it still takes a few more epochs to learn the information thoroughly. to summarise, our proposed entropy measure can generally approximate the level of information contained in the embeddings. After examining the validity of our proposed measure, we can compare the entropy of BPR and BPR_{DNS} shown as green and orange lines,

Table 3: Performances of two advanced NS-based baselines and four general recommenders with their DNS variants on the three used datasets. All metrics are computed at rank cutoff 10. The best result is highlighted in bold and the largest percentage improvement is underlined; * denotes a significant difference between the performance of a DNS variant and that of the baseline, while † denotes a significant difference among the four DNS-based models, according to the paired t-test with the Holm-Bonferroni correction for p <0.01.

	MovieLens			Yelp			Amazon		
	NDCG	Recall	MAP	NDCG	Recall	MAP	NDCG	Recall	MAP
PRIS	0.3298*	0.2301*	0.1105*	0.2205*	0.2897*	0.1214*	0.2578*	0.3806*	0.1970*
MixGCF	0.3300*	0.2351*	0.1179*	0.2100*	0.2705*	0.1104*	0.2308*	0.3698*	0.1806*
BPR	0.2805*	0.1917*	0.0886*	0.1230*	0.1683*	0.0600*	0.0834*	0.1632*	0.0526*
BPR_{DNS}	0.3394†	0.2281†	0.1090†	0.1513†	0.2019†	0.0726†	0.1092†	0.2089†	0.0678†
Improvement	20.9%	19.0%	23.0%	23.0%	20.1%	21.0%	30.9%	28.0%	28.9%
LightGCN	0.3425*	0.2504*	0.1252*	0.2285*	0.3067*	0.1255*	0.2599*	0.4111*	0.1980*
$LightGCN_{DNS}$	0.3938†	0.2905†	0.1488†	0.2696†	0.3690†	**0.1472**†	0.3119†	0.4805†	**0.2356**†
Improvement	15.0%	16.1%	18.8%	18.0%	20.3%	17.3%	20.0%	16.9%	18.9%
SGL	0.3575*	0.2716*	0.1366*	0.2198*	0.2867*	0.1204*	0.2664*	0.3983*	0.2092*
SGL_{DNS}	**0.3942**†	**0.3042**†	**0.1537**†	0.2528†	0.3340†	0.1360†	0.3143†	0.4680†	0.2489†
Improvement	10.3%	12.0%	12.5%	15.1%	16.5%	12.9%	18.1%	17.5%	19.0%
SimGCL	0.3505*	0.2689*	0.1301*	0.2328*	0.3278*	0.1298*	0.2754*	0.4229*	0.2187*
$SimGCL_{DNS}$	0.3870†	0.2998†	0.1496†	**0.2707**†	**0.3731**†	0.1467†	**0.3175**†	**0.4910**†	**0.2524**†
Improvement	10.4%	11.5%	15.0%	16.3%	13.5%	13.0%	15.3%	16.1%	15.4%

respectively, in Figure 3. The figure clearly shows that the contrastive negative items provided by our DNS scheme continuously contain more information than the random negative items when using the entropy measure. Furthermore, a smoother entropy line of BPR_{DNS} (green) indicates that our DNS scheme also provides a more constant information input, which is beneficial for a faster convergence as suggested in [59].

In answer to **RQ4**, we can conclude that our proposed entropy measure can generally approximate the level of information contained in the embeddings. Finally, our results show that contrastive negative items sampled by our proposed DNS scheme indeed provide a higher information gain than those random negative items.

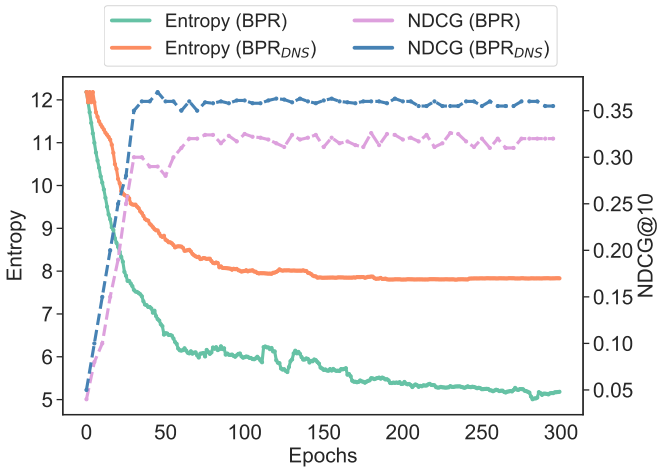

Figure 3: Entropy comparisons of BPR_{DNS} and BPR on the MovieLens 1M dataset.

6 CONCLUSIONS

In this paper, we proposed DNS, a dynamic negative sampling scheme for contrastive recommender systems (RSs), which can be deployed in many existing general RSs to improve their performances. In particular, DNS can dynamically sample contrastive negative items at each checkpoint solely based on the cosine similarity scores between the users and items' embeddings without using any auxiliary information. Besides, we proposed a contrastive objective function to leverage these additionally sampled negative items. This objective function is able to enhance the performance of general recommenders. We also proposed an entropy measure tailored for RSs so as to estimate the information contained in the embeddings. Our extensive evaluation of the DNS-based models on three datasets in comparison to six baselines showed that our proposed scheme can generally improve the performance of four general recommenders and can significantly outperform two strong baselines using different advanced negative sampling approaches. Specifically, our DNS scheme can improve the performance of a general recommender by up-to 30.9% in terms of the NDCG@10 metric (see Table 3). In addition, using DNS to provide contrastive negative items can bring a continuously higher and more constant information gain (see Figure 3). The significant performance improvements obtained by our proposed DNS scheme suggest that DNS is a superior scheme over other existing pairwise learning schemes. As future work, we aim to devise an integrated scheme that can automatically decide how many negative items to retrieve and how frequently these dynamic negative items should be updated. We will also explore how to conceive such an integrated scheme to explicitly maximise the entropy between the users' embeddings and the items' embeddings.

REFERENCES

[1] Alain, G., Lamb, A., Sankar, C., Courville, A., Bengio, Y.: Variance reduction in sgd by distributed importance sampling. In: Proceedings of ICLR (2016)

[2] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D.A., Hernández, M.V., Wardlaw, J., Rueckert, D.: Gan augmentation: Augmenting training data using generative adversarial networks. arXiv preprint arXiv:1810.10863 (2018)

[3] Carter, T.: An introduction to information theory and entropy. Complex systems summer school, Santa Fe (2007)

[4] Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., He, X.: Bias and debias in recommender system: A survey and future directions. arXiv preprint arXiv:2010.03240 (2020)

[5] Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of ICML (2020)

[6] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.: Big self-supervised models are strong semi-supervised learners. In: Proceedings of NeurIPS (2020)

[7] Chuang, C.Y., Robinson, J., Yen-Chen, L., Torralba, A., Jegelka, S.: Debiased contrastive learning. In: Proceedings of NeurIPS (2020)

[8] Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL (2019)

[9] Ding, J., Quan, Y., He, X., Li, Y., Jin, D.: Reinforced negative sampling for recommendation with exposure data. In: Proceedings of IJCAI (2019)

[10] Eksombatchai, C., Jindal, P., Liu, J.Z., Liu, Y., Sharma, R., Sugnet, C., Ulrich, M., Leskovec, J.: Pixie: A system for recommending 3+ billion items to 200+ million users in real-time. In: Proceedings of WWW (2018)

[11] Fikir, O.B., Yaz, I.O., Özyer, T.: A movie rating prediction algorithm with collaborative filtering. In: Proceedings of ICASNAM (2010)

[12] Gantner, Z., Drumond, L., Freudenthaler, C., Rendle, S., Schmidt-Thieme, L.: Learning attribute-to-feature mappings for cold-start recommendations. In: Proceedings of ICDM (2010)

[13] Gomez-Uribe, C.A., Hunt, N.: The Netflix recommender system: Algorithms, business value, and innovation. ACM Transactions on Management Information Systems **6**(4), 1–19 (2015)

[14] Hariadi, A.I., Nurjanah, D.: Hybrid attribute and personality based recommender system for book recommendation. In: Proceedings of ICoDSE (2017)

[15] He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: Simplifying and powering graph convolution network for recommendation. In: Proceedings of SIGIR (2020)

[16] He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: Proceedings of WWW (2017)

[17] Hershey, J.R., Olsen, P.A.: Approximating the Kullback Leibler divergence between Gaussian mixture models. In: Proceedings of ICASSP (2007)

[18] Huang, T., Dong, Y., Ding, M., Yang, Z., Feng, W., Wang, X., Tang, J.: Mixgcf: An improved training method for graph neural network-based recommender systems. In: Proceedings of SIGKDD (2021)

[19] Johnson, T.B., Guestrin, C.: Training deep models faster with robust, approximate importance sampling. In: Proceedings of NeurIPS (2018)

[20] Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., Yih, W.T.: Dense passage retrieval for open-domain question answering. In: Proceedings of EMNLP (2020)

[21] Katharopoulos, A., Fleuret, F.: Not all samples are created equal: Deep learning with importance sampling. In: Proceeings of ICML (2018)

[22] Khenissi, S., Mariem, B., Nasraoui, O.: Theoretical modeling of the iterative properties of user discovery in a collaborative filtering recommender system. In: Proceedings of RecSys (2020)

[23] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. In: Proceedings of NeurIPS (2020)

[24] Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Physical review E **69**(6) (2004)

[25] Krichene, W., Rendle, S.: On sampled metrics for item recommendation. In: Proceedings of SIGKDD (2020)

[26] Kywe, S.M., Lim, E.P., Zhu, F.: A survey of recommender systems in Twitter. In: Proceedings of ICSI (2012)

[27] Leino, J., Räihä, K.J.: Case amazon: ratings and reviews as part of recommendations. In: Proceedings of RecSys (2007)

[28] Lian, D., Liu, Q., Chen, E.: Personalized ranking with importance sampling. In: Proceedings of WWW (2020)

[29] Liu, J., Wu, C., Xiong, Y., Liu, W.: List-wise probabilistic matrix factorization for recommendation. Information Sciences **278**, 434–447 (2014)

[30] Liu, Z., Ma, Y., Ouyang, Y., Xiong, Z.: Contrastive learning for recommender system. arXiv preprint arXiv:2101.01317 (2021)

[31] Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. Decision Support Systems **74**, 12–32 (2015)

[32] Manotumruksa, J., Macdonald, C., Ounis, I.: A personalised ranking framework with multiple sampling criteria for venue recommendation. In: Proceedings of CIKM (2017)

[33] Martinez, M., Stiefelhagen, R.: Taming the cross entropy loss. In: Proceedings of DAGM (2018)

[34] Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)

[35] Park, D.H., Chang, Y.: Adversarial sampling and training for semi-supervised information retrieval. In: Proceedings of WWW (2019)

[36] Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: Proceedings of UAI (2009)

[37] Rendle, S., Krichene, W., Zhang, L., Anderson, J.: Neural collaborative filtering vs. matrix factorization revisited. In: Proceedings of RecSys (2020)

[38] Robinson, J.D., Chuang, C.Y., Sra, S., Jegelka, S.: Contrastive learning with hard negative samples. In: Proceedings of ICLR (2020)

[39] Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., Isola, P.: What makes for good views for contrastive learning? arXiv preprint arXiv:2005.10243 (2020)

[40] Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)

[41] Tran, V.A., Hennequin, R., Royo-Letelier, J., Moussallam, M.: Improving collaborative metric learning with efficient negative sampling. In: Proceedings of SIGIR (2019)

[42] Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B., Lee, D.L.: Billion-scale commodity embedding for e-commerce recommendation in alibaba. In: Proceedings of SIGKDD (2018)

[43] Wang, J., De Vries, A.P., Reinders, M.J.: Unified relevance models for rating prediction in collaborative filtering. ACM Transactions on Information Systems **26**(3) (2008)

[44] Wang, J., Yu, L., Zhang, W., Gong, Y., Xu, Y., Wang, B., Zhang, P., Zhang, D.: Irgan: A minimax game for unifying generative and discriminative information retrieval models. In: Proceedings of SIGIR (2017)

[45] Wang, X., Ounis, I., Macdonald, C.: Leveraging review properties for effective recommendation. In: Proceedings WWW (2021)

[46] Wang, X., He, X., Wang, M., Feng, F., Chua, T.S.: Neural graph collaborative filtering. In: Proceedings of SIGIR (2019)

[47] Wang, X., Xu, Y., He, X., Cao, Y., Wang, M., Chua, T.S.: Reinforced negative sampling over knowledge graph for recommendation. In: Proceedings of WWW (2020)

[48] Wu, F., Zhang, T., Souza Jr, A.H.d., Fifty, C., Yu, T., Weinberger, K.Q.: Simplifying graph convolutional networks. In: Proceedings of ICML (2019)

[49] Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., Xie, X.: Self-supervised graph learning for recommendation. In: Proceedings of SIGIR (2021)

[50] Wu, Z., Wang, S., Gu, J., Khabsa, M., Sun, F., Ma, H.: Clear: Contrastive learning for sentence representation. arXiv preprint arXiv:2012.15466 (2020)

[51] Xiong, L., Xiong, C., Li, Y., Tang, K.F., Liu, J., Bennett, P., Ahmed, J., Overwijk, A.: Approximate nearest neighbor negative contrastive learning for dense text retrieval. In: Proceedings of ICLR (2020)

[52] Yang, X., Li, Y., Luo, J.: Pinterest board recommendation for Twitter users. In: Proceedings of SIGMM (2015)

[53] Yang, Z., Ding, M., Zou, X., Tang, J., Xu, B., Zhou, C., Yang, H.: Region or global a principle for negative sampling in graph-based recommendation. IEEE Transactions on Knowledge and Data Engineering (2022)

[54] Yih, W.t., Toutanova, K., Platt, J.C., Meek, C.: Learning discriminative projections for text similarity measures. In: Proceedings of CoNLL (2011)

[55] Yu, J., Yin, H., Li, J., Gao, M., Huang, Z., Cui, L.: Enhance social recommendation with adversarial graph convolutional networks. IEEE Transactions on Knowledge and Data Engineering (2020)

[56] Yu, J., Yin, H., Xia, X., Chen, T., Cui, L., Hung, N.Q.V.: Are graph augmentations necessary? simple graph contrastive learning for recommendation. In: Proceedings of SIGIR (2022)

[57] Zhang, W., Chen, T., Wang, J., Yu, Y.: Optimizing top-n collaborative filtering via dynamic negative item sampling. In: Proceedings of SIGIR (2013)

[58] Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: Proceedings of NeurIPS (2018)

[59] Zhao, S., Sinha, A., He, Y., Perreault, A., Song, J., Ermon, S.: Comparing distributions by measuring differences that affect decision making. In: Proceedings of ICLR (2021)

[60] Zhao, X., Zhang, L., Xia, L., Ding, Z., Yin, D., Tang, J.: Deep reinforcement learning for list-wise recommendations. arXiv preprint arXiv:1801.00209 (2017)

[61] Zheng, Y., Gao, C., Li, X., He, X., Li, Y., Jin, D.: Disentangling user interest and popularity bias for recommendation with causal embedding. arXiv preprint arXiv:2006.11011 (2020)

[62] Zhou, C., Ma, J., Zhang, J., Zhou, J., Yang, H.: Contrastive learning for debiased candidate generation in large-scale recommender systems. In: Proceedings of SIGKDD (2021)